Hardware and Software for Approximate Computing

Correctness is a fundamental tenet in computer systems. In many cases, however, perfect answers are unnecessary or even impossible: small errors can be acceptable in applications such as vision, machine learning, speech recognition, search, graphics, and physical simulation. Approximate computing is a new research direction that improves efficiency by carefully relaxing correctness constraints. My research seeks to get the most out of approximate computing using collaboration between software and hardware. In hardware, I have proposed CPU designs, accelerators, and storage systems that can gracefully trade off quality for efficiency. In software, I have designed type systems, debugging tools, compilers, and a language construct called a probabilistic assertion for controlling approximation's impact. Together, software and hardware for approximate computing can exploit applications' latent resiliency to yield new performance and energy benefits.

Bio:
Adrian Sampson is a Ph.D. candidate the University of Washington, where his advisors are Luis Ceze and Dan Grossman. His research combines computer architecture, programming languages, and compilers. His work on approximate computing has been supported by fellowships from Facebook, Qualcomm, and Google.